Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A system for organizing and accessing a database, the system comprising:

~~a primary B+tree index;~~

~~a secondary B+tree index;~~

~~a plurality of mapping table row identifiers stored in the secondary B+tree index, where each mapping row identifier is an entry in a row of the secondary B+tree index and uniquely identifies a row within a mapping table;~~

~~a first plurality of database addresses for leaf blocks of the primary B+tree index stored in the secondary B+tree index; and~~

~~a corresponding number of database addresses for leaf blocks of the primary B+tree index stored in a row of the mapping table, wherein data stored within the primary B+tree is indexed-organized and accessed using the secondary B+tree or mapping table;~~

~~wherein the database addresses correspond to guess-database addresses, where each ,guess-database address is a guess as to what address block of the primary B+tree a row may be found.~~

a secondary index for a primary B+tree, wherein the secondary index comprises a plurality of rows each comprising an index key value, and a guess-database address value that represents a guess as to an address block of the primary B+tree where a row may be found, where data stored in the database is retrieved using the secondary index for the primary B+tree.

2. (currently amended) The system according to claim 1, wherein the guess-database address values are 4 bytes of ~~the database addresses~~ address blocks in the primary B+tree ~~are stored in the secondary B+tree index~~.

3

3. (cancelled)

4. (currently amended) The system according to claim 1, further comprising:

a guess-database address quality statistic for the secondary index ~~and a guess-database~~
~~address quality statistic for the mapping table~~, where ~~each of~~ the quess-database address quality
statistic represents a ratio of how often the guesses as to where rows may be found in an address
block of the primary B+tree are accurate.

5. (previously presented) A method for managing a database system, the method
comprising:

creating a secondary index for a primary B+tree ~~structure~~, wherein the secondary index
~~structure~~ comprises a plurality of rows each comprising an index key value, ~~a mapping table~~
~~rowid value that uniquely identifies a row within a mapping table,~~ and a guess-database address
value that represents a guess as to an address block of ~~a~~ the primary B+tree where a row may be
found; and

retrieving data stored in the database system using the secondary index for ~~a~~ the primary
B+tree.

6. (currently amended) The method according to claim 18 [[5]], further comprising:

inserting a row of the secondary index ~~structure~~, wherein inserting the row comprises
inserting a row comprising an index key value, a mapping table rowid value and a guess database
address value.

7. (currently amended) The method according to claim 18 [[5]], further comprising:

deleting a row of the secondary index, wherein deleting the row comprises locating a row comprising an index key value and a mapping table rowid value ~~row identifier~~ and deleting the row.

8. (currently amended) The method according to claim 18 [[5]], further comprising:

updating the secondary index, wherein updating the secondary index comprises locating a row of the secondary index comprising an old index key value and a mapping table rowid value ~~row identifier~~,

deleting the row and inserting in the row a new index key value, a mapping table rowid value ~~row identifier~~ and a guess database address value.

9. (currently amended) The method according to claim 18 [[5]], wherein retrieving data stored in the database system further comprises: ~~comprising carrying out a query utilizing the secondary index, wherein carrying out the query comprises:~~

~~utilizing a guess datab,ase address stored as part of a secondary index row to find a row in the primary B+tree structure;~~

obtaining a first guess database address value representing a first address block of the primary B+tree structure ~~a target database block from the row in the primary B+tree structure~~;

searching the first address ~~target database~~ block of the primary B+tree for a row that contains a mapping table rowid value ~~row identifier~~ that is the same as a mapping table rowid value ~~row identifier~~ in the row where the first guess database address value is stored in the secondary index row; and

if the mapping table rowid is found ~~the row in the database block matches the target database block~~, then the correct row in the primary B+tree ~~database~~ has been located and the data is retrieved ~~and the query is completed~~.

5

10. (currently amended) The method according to claim 9, <u>wherein retrieving data stored in the</u> database system <u>further comprises:</u>

<u>if the mapping table rowid value is not found, then</u> ~~wherein if the row in the database block does not match the target database block, carrying out the query further comprises:~~ <u>utilizing the mapping table rowid value stored in the row where the first guess database address value is stored in the secondary index to access</u> ~~accessing the~~ <u>a</u> mapping table row stored in <u>a mapping table</u> ~~the secondary index row;~~

utilizing a <u>second</u> guess-database address <u>value</u> stored in the mapping table row to access <u>a</u> second address ~~a target~~ block of the <u>primary B+tree</u> ~~database~~;

searching the <u>second address</u> ~~a target~~ block <u>of the primary B+tree</u> for a primary key that matches a primary key stored in the mapping table row; and

if the primary key is found, then the <u>data is retrieved</u> ~~query is completed~~.

11. (currently amended) The method according to claim 10, wherein if the primary key is not ~~located~~ <u>found</u> <u>retrieving data</u> ~~carrying out the query~~ further comprises:

traversing the primary B+tree ~~structure~~ utilizing the primary key value from the mapping table row to identify the database address to complete the query.

12. (previously presented) The method according to claim 11, further comprising:

maintaining a guess-database address quality statistic for the secondary index;

maintaining a guess-database address quality statistic for the mapping table;

utilizing the statistics to assess guess-database address quality; and

carrying out the query based upon guess-database quality in the secondary index and mapping table, where each of the quess-database address quality statistic represents a ratio of how

often the guesses as to where rows may be found in an address block of the primary B+tree are accurate.

13. (original) The method according to claim 12, further comprising:

estimating guess-database address quality;

estimating the cost of the query based upon the estimated guess-database address quality; and

carrying out the query starting with an index structure with the lowest estimated cost.

14. (currently amended) The method according to claim 5, wherein the guess-database address values are ~~only~~ 4 bytes of address blocks in the primary B+tree ~~the guess database address value are stored in the secondary index row~~.

15. (currently amended) A computer program product for performing a process of managing a database system, comprising:

a computer readable medium; and

computer program instructions, recorded on the computer readable medium, executable by a processor, for performing the steps of:

creating a secondary index for a primary B+tree ~~structure~~, wherein the secondary index ~~structure~~ comprises a plurality of rows each comprising an index key value, a mapping table rowid value that uniquely identifies a row within a mapping table, and a guess-database address value that represents a guess as to an address block of a the primary B+tree where a row may be found; and

retrieving data stored in the database system using the secondary index for a the primary B+tree.

16. (currently amended) A system for performing a database management process, comprising

a processor operable to execute computer program instructions; and

a memory operable to store computer program instructions executable by the processor, for performing the steps of:

creating a secondary index for a primary B+tree ~~structure~~, wherein the secondary index ~~structure~~ comprises a plurality of rows each comprising an index key value, ~~a mapping table rowid value that uniquely identifies a row within a mapping table~~, and a guess-database address value that represents a guess as to an address block of a the primary B+tree where a row may be found; and

retrieving data stored in the database system using the secondary index for a the primary B+tree.

17. (new) The system according to claim 1, wherein each row in the plurality of rows further comprising a mapping table rowid value that identifies a row within a mapping table.

18. (new) The method according to claim 5, wherein each row in the plurality of rows further comprising a mapping table rowid value that identifies a row within a mapping table.

19. (new) The computer program product according to claim 15, wherein each row in the plurality of rows further comprising a mapping table rowid value that identifies a row within a mapping table.

20. (new) The system according to claim 16, wherein each row in the plurality of rows further comprising a mapping table rowid value that identifies a row within a mapping table.